```
R version 4.2.2 (2022-10-31 ucrt) -- "Innocent and Trusting"
Copyright (C) 2022 The R Foundation for Statistical Computing
Platform: x86_64-w64-mingw32/x64 (64-bit)

R is free software and comes with ABSOLUTELY NO WARRANTY.
You are welcome to redistribute it under certain conditions.
Type 'license()' or 'licence()' for distribution details.

  Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

[Previously saved workspace restored]

> # Define function Zinterval
> Zinterval <- function(n, lambda) {
+   A <- rexp(n,lambda)
+   lower <- mean(x) - qnorm(0.975) * sd(A) / sqrt(n)
+   upper <- mean(x) + qnorm(0.975) * sd(A) / sqrt(n)
+   Tmean = 1/lambda
+   if(upper>Tmean & lower<Tmean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function zproportion
> zproportion <- function(n, lambda) {
+   CI <- replicate(5000, Zinterval(n, lambda))
+   value <- CI[which (Present_in_CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
Error in if (upper > Tmean & lower < Tmean) { :
  missing value where TRUE/FALSE needed
In addition: Warning messages:
1: In mean.default(x) : argument is not numeric or logical: returning NA
2: In mean.default(x) : argument is not numeric or logical: returning NA
>
> # Define function mean.star
> Mean_star<- function(n,lambda) {
+   U_star <- rexp(n, lambda)
+   return (mean(U_star))
+ }
>
> # Define function boot_CI
> boot_CI <- function(n, lambda) {
+   B <- rexp(n,lambda)
+   True_Mean <- 1/lambda
+   lambda1 = 1/mean(B)
+   C <- replicate(1000, mean_star(n, lambda1))
+   bound <- sort(C)[c(25, 975)]
+   if(bound[2]>True_Mean & bound[1]<True_Mean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
```

```
+ }
>
> # Define function bproportion
> bproportion <- function(n, lambda) {
+   CI <- replicate(5000, boot_CI(n, lambda))
+   value <-CI[which (Present_in_CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for bproportion
> bproportion(5,0.01)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
> # Generate proportion values for bootstrap and z-interval for all combinations of n and lambda
>
> Z_matrix <- matrix(c(zproportion(5,0.01), zproportion(10,0.01), zproportion(30,0.01), zproporti
on(100,0.01), zproportion(5,0.1), zproportion(10,0.1), zproportion(30,0.1), zproportion(100,0.1),
 zproportion(5,1), zproportion(10,1), zproportion(30,1), zproportion(100,1), zproportion(5,10), z
proportion(10,10), zproportion(30,10), zproportion(100,10)),nrow=4,ncol =4)
Error in if (upper > Tmean & lower < Tmean) { :
  missing value where TRUE/FALSE needed
In addition: Warning messages:
1: In mean.default(x) : argument is not numeric or logical: returning NA
2: In mean.default(x) : argument is not numeric or logical: returning NA
>
>
> B_matrix <- matrix(c(bproportion(5,0.01), bproportion(10,0.01), bproportion(30,0.01), bproporti
on(100,0.01), bproportion(5,0.1), bproportion(10,0.1), bproportion(30,0.1), bproportion(100,0.1),
 bproportion(5,1), bproportion(10,1), bproportion(30,1), bproportion(100,1), bproportion(5,10), b
proportion(10,10), bproportion(30,10), bproportion(100,10)),nrow=4,ncol =4)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
> par(mfrow=c(2,2))
> plot(c(5,10,30,100), Z_matrix[,1], main = "L = 0.01", xlab = 'n', ylab ='Proportions', col = 'r
ed',
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
Error in xy.coords(x, y, xlabel, ylabel, log) :
  object 'Z_matrix' not found
> lines(c(5,10,30,100), B_matrix[,1], col = 'blue', type = 'b')
Error in xy.coords(x, y) : object 'B_matrix' not found
> # Define function Zinterval
> Zinterval <- function(n, lambda) {
+   A <- rexp(n,lambda)
+   lower <- mean(A) - qnorm(0.975) * sd(A) / sqrt(n)
+   upper <- mean(A) + qnorm(0.975) * sd(A) / sqrt(n)
+   Tmean = 1/lambda
+   if(upper>Tmean & lower<Tmean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function zproportion
> zproportion <- function(n, lambda) {
+   CI <- replicate(5000, Zinterval(n, lambda))
+   value <- CI[which (Present_in_CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
Error in which(Present_in_CI == 1) : object 'Present_in_CI' not found
>
> # Define function mean.star
> Mean_star<- function(n,lambda) {
+   U_star <- rexp(n, lambda)
+   return (mean(U_star))
```

```
+ }
>
> # Define function boot_CI
> boot_CI <- function(n, lambda) {
+   B <- rexp(n,lambda)
+   True_Mean <- 1/lambda
+   lambda1 = 1/mean(B)
+   C <- replicate(1000, mean_star(n, lambda1))
+   bound <- sort(C)[c(25, 975)]
+   if(bound[2]>True_Mean & bound[1]<True_Mean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function bproportion
> bproportion <- function(n, lambda) {
+   CI <- replicate(5000, boot_CI(n, lambda))
+   value <-CI[which (Present_in_CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for bproportion
> bproportion(5,0.01)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
> # Generate proportion values for bootstrap and z-interval for all combinations of n and lambda
>
> Z_matrix <- matrix(c(zproportion(5,0.01), zproportion(10,0.01), zproportion(30,0.01), zproporti
on(100,0.01), zproportion(5,0.1), zproportion(10,0.1), zproportion(30,0.1), zproportion(100,0.1),
 zproportion(5,1), zproportion(10,1), zproportion(30,1), zproportion(100,1), zproportion(5,10), z
proportion(10,10), zproportion(30,10), zproportion(100,10)),nrow=4,ncol =4)
Error in which(Present_in_CI == 1) : object 'Present_in_CI' not found
>
>
> B_matrix <- matrix(c(bproportion(5,0.01), bproportion(10,0.01), bproportion(30,0.01), bproporti
on(100,0.01), bproportion(5,0.1), bproportion(10,0.1), bproportion(30,0.1), bproportion(100,0.1),
 bproportion(5,1), bproportion(10,1), bproportion(30,1), bproportion(100,1), bproportion(5,10), b
proportion(10,10), bproportion(30,10), bproportion(100,10)),nrow=4,ncol =4)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
> par(mfrow=c(2,2))
> plot(c(5,10,30,100), Z_matrix[,1], main = "L = 0.01", xlab = 'n', ylab ='Proportions', col = 'r
ed',
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
Error in xy.coords(x, y, xlabel, ylabel, log) :
  object 'Z_matrix' not found
> lines(c(5,10,30,100), B_matrix[,1], col = 'blue', type = 'b')
Error in xy.coords(x, y) : object 'B_matrix' not found
>
>
>
>
> # Define function Zinterval
> Zinterval <- function(n, lambda) {
+   A <- rexp(n,lambda)
+   lower <- mean(A) - qnorm(0.975) * sd(A) / sqrt(n)
+   upper <- mean(A) + qnorm(0.975) * sd(A) / sqrt(n)
+   Tmean = 1/lambda
+   if(upper>Tmean & lower<Tmean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
```

```
> # Define function zproportion
> zproportion <- function(n, lambda) {
+   CI <- replicate(5000, Zinterval(n, lambda))
+   value <- CI[which (CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
[1] 0.8182
>
> # Define function mean.star
> Mean_star<- function(n,lambda) {
+   U_star <- rexp(n, lambda)
+   return (mean(U_star))
+ }
>
> # Define function boot_CI
> boot_CI <- function(n, lambda) {
+   B <- rexp(n,lambda)
+   True_Mean <- 1/lambda
+   lambda1 = 1/mean(B)
+   C <- replicate(1000, mean_star(n, lambda1))
+   bound <- sort(C)[c(25, 975)]
+   if(bound[2]>True_Mean & bound[1]<True_Mean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function bproportion
> bproportion <- function(n, lambda) {
+   CI <- replicate(5000, boot_CI(n, lambda))
+   value <-CI[which (CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for bproportion
> bproportion(5,0.01)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
> # Generate proportion values for bootstrap and z-interval for all combinations of n and lambda
>
> Z_matrix <- matrix(c(zproportion(5,0.01), zproportion(10,0.01), zproportion(30,0.01), zproporti
on(100,0.01), zproportion(5,0.1), zproportion(10,0.1), zproportion(30,0.1), zproportion(100,0.1),
 zproportion(5,1), zproportion(10,1), zproportion(30,1), zproportion(100,1), zproportion(5,10), z
proportion(10,10), zproportion(30,10), zproportion(100,10)),nrow=4,ncol =4)
>
>
> B_matrix <- matrix(c(bproportion(5,0.01), bproportion(10,0.01), bproportion(30,0.01), bproporti
on(100,0.01), bproportion(5,0.1), bproportion(10,0.1), bproportion(30,0.1), bproportion(100,0.1),
 bproportion(5,1), bproportion(10,1), bproportion(30,1), bproportion(100,1), bproportion(5,10), b
proportion(10,10), bproportion(30,10), bproportion(100,10)),nrow=4,ncol =4)
Error in mean_star(n, lambda1) : could not find function "mean_star"
>
>
>
>
>
>
> # Define function Zinterval
> Zinterval <- function(n, lambda) {
+   A <- rexp(n,lambda)
+   lower <- mean(A) - qnorm(0.975) * sd(A) / sqrt(n)
+   upper <- mean(A) + qnorm(0.975) * sd(A) / sqrt(n)
+   Tmean = 1/lambda
+   if(upper>Tmean & lower<Tmean) {
```

```
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function zproportion
> zproportion <- function(n, lambda) {
+   CI <- replicate(5000, Zinterval(n, lambda))
+   value <- CI[which (CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for zproportion
> zproportion(5,0.01)
[1] 0.815
>
> # Define function mean.star
> Mean_star<- function(n,lambda) {
+   U_star <- rexp(n, lambda)
+   return (mean(U_star))
+ }
>
> # Define function boot_CI
> boot_CI <- function(n, lambda) {
+   B <- rexp(n,lambda)
+   True_Mean <- 1/lambda
+   lambda1 = 1/mean(B)
+   C <- replicate(1000, Mean_star(n, lambda1))
+   bound <- sort(C)[c(25, 975)]
+   if(bound[2]>True_Mean & bound[1]<True_Mean) {
+     return (1)
+   }
+   else {
+     return (0)
+   }
+ }
>
> # Define function bproportion
> bproportion <- function(n, lambda) {
+   CI <- replicate(5000, boot_CI(n, lambda))
+   value <-CI[which (CI == 1)]
+   return (length(value)/5000)
+ }
>
> # Get value of n = 5 and lambda = 0.01 for bproportion
> bproportion(5,0.01)
[1] 0.8994
>
> # Generate proportion values for bootstrap and z-interval for all combinations of n and lambda
>
> Z_matrix <- matrix(c(zproportion(5,0.01), zproportion(10,0.01), zproportion(30,0.01), zproporti
on(100,0.01), zproportion(5,0.1), zproportion(10,0.1), zproportion(30,0.1), zproportion(100,0.1),
 zproportion(5,1), zproportion(10,1), zproportion(30,1), zproportion(100,1), zproportion(5,10), z
proportion(10,10), zproportion(30,10), zproportion(100,10)),nrow=4,ncol =4)
>
>
> B_matrix <- matrix(c(bproportion(5,0.01), bproportion(10,0.01), bproportion(30,0.01), bproporti
on(100,0.01), bproportion(5,0.1), bproportion(10,0.1), bproportion(30,0.1), bproportion(100,0.1),
 bproportion(5,1), bproportion(10,1), bproportion(30,1), bproportion(100,1), bproportion(5,10), b
proportion(10,10), bproportion(30,10), bproportion(100,10)),nrow=4,ncol =4)
> par(mfrow=c(2,2))
> plot(c(5,10,30,100), Z_matrix[,1], main = "L = 0.01", xlab = 'n', ylab ='Proportions', col = 'r
ed',
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), B_matrix[,1], col = 'blue', type = 'b')
> plot(c(5,10,30,100), Z_matrix[,2], main = "L = 0.1", xlab = 'n', ylab = 'Proportions', col ='re
d',
```

```
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), B_matrix[,2], col = 'blue', type = 'b')
>
> plot(c(5,10,30,100), Z_matrix[,3], main = "L = 1", xlab = 'n', ylab = 'Proportions', col = 'red
',
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), B_matrix[,3], col = 'blue', type = 'b')
>
> plot(c(5,10,30,100), Z_matrix[,4], main = "L = 10", xlab = 'n', ylab = 'Proportions', col = 're
d',
+ type = 'b', xlim = c(1,100), ylim = c(0,1))
> lines(c(5,10,30,100), B_matrix[,4], col = 'blue', type = 'b')
>
>
> plot(c(0.01,0.1,1,10), Z_matrix[1,], main = "N = 5", xlab = 'Lambda', ylab = 'Proportions', col
 ='red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), B_matrix[1,], col = 'blue', type = 'b')
>
> plot(c(0.01,0.1,1,10), Z_matrix[2,], main = "N = 10", xlab = 'Lambda', ylab = 'Proportions', co
l ='red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
>  lines(c(0.01,0.1,1,10), B_matrix[2,], col = 'blue', type = 'b')
> plot(c(0.01,0.1,1,10), Z_matrix[3,], main = "N = 30", xlab = 'Lambda', ylab = 'Proportions', co
l ='red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), B_matrix[3,], col = 'blue', type = 'b')
>
> plot(c(0.01,0.1,1,10), Z_matrix[4,], main = "N = 100", xlab = 'Lambda', ylab ='Proportions', co
l ='red', type = 'b', xlim = c(0.01,10), ylim = c(0,1))
> lines(c(0.01,0.1,1,10), B_matrix[4,], col = 'blue', type = 'b')
>
>
> z_matrix
Error: object 'z_matrix' not found
> Z_matrix
       [,1]   [,2]   [,3]   [,4]
[1,] 0.8182 0.8132 0.8124 0.8064
[2,] 0.8706 0.8640 0.8706 0.8674
[3,] 0.9094 0.9148 0.9212 0.9188
[4,] 0.9360 0.9370 0.9372 0.9414
> B_matrix
       [,1]   [,2]   [,3]   [,4]
[1,] 0.8924 0.8930 0.8914 0.8998
[2,] 0.9194 0.9204 0.9236 0.9212
[3,] 0.9342 0.9310 0.9390 0.9428
[4,] 0.9408 0.9500 0.9408 0.9476
>
```